**Primary TdBench Commands**

**# comment**

Documents scripts. (Blank lines are ignored).

[**?**](#_Meet_Your_Coach.)

Request current in-progress setup of a test and get suggestions of what else is needed

**help [ topic | command | ? ]**

Get details on the syntax and usage of a command

**define [test-name] {test description}**

Define name and description of a test.

**queue [queue-name] [query; | "OS" command; | pathname | wildcard pathname | "kill | stop" {".all"};}**

Specify a queue of commands or queries/subqueries for the test. Follow queries with ; or it will be taken as file name.

**worker [queue-name] db-alias-name {count} | “OS” {count}**

Defines workers to run the queue commands.

**run { # [“h”|”m”|”s” ] | serial {kill}.}**

Starts the test. Example: **run 30m** … executes for 30 minutes

**quit**

Exits TdBench

**Environment/Script Control Commands**

**cd [directory]**

Change TDBench working directory.

**class [class-name-of-JDBC-driver] [JDBC Protocol e.g. "jdbc:teradata"] [file-name(s)-of-JDBC-driver-jar]**

Load JDBC driver class from a file.

**db [alias-name] [JDBC Protocol e.g. “jdbc:teradata://192.168.1.28/database=my\_benchmark”] {username {password}}**

Define a database alias for use in WORKER and/or SQL commands.

**color [input] [output] [error]**

sets ANSI screen colors

**echo {text:filename | append:filename} {comments + variables} {delim=*string\_on\_line\_later*}**

Displays text on console or writes text (with variables) to a file. Can be multi-line with delim=

**exec [filename|pathname] {argument-1 ...}**

Alias for INCLUDE. Read TDBench commands from a file (can be nested and allows parameters :i1, :i2 …)

**include [filename|pathname] {argument-1 ...}**

Read TDBench commands from a file (can be nested and allows parameters :i1, :i2 …)

**Read [variable] {prompt}**

Ask user for value to use in a script

**set [variable] {value}**

sets or clears contents of a variable

**sleep { # {“s" | "m" | "h"} } | {yyyy-mm-dd-}hh:mm{:ss}**

Will go to sleep until a time or a relative time has passed.

**status [# seconds | 0]**

Will display queue status during test (versus all results.)

**trace [on | off | list | save filename | clear]**

Controls echo of statements to console, lists out or saves them to file. Example:  **trace list** … will list command entered

**zip [work-directory-name] [zip-directory-name] {erase}**

Defines a directory to archive using zip protocol after test.

**Commands to Insert Commands During Test**

**before\_run ["SQL" ... sql command; | "OS" ...os command; | "DELETE" ]**

Statements to be run before RUN statement. (allows substitution of TdBench variables)

**after\_run ["SQL" ... sql command; | "OS" ...os command; | "DELETE" ]**

Statements to be run after the test completes

**before\_worker [queue-name] [query; | "OS" command; | pathname]**

Specify one or more queries to run in each worker before a queue. (e.g. database command)

**before\_query [queue-name] [query;]**

Specify a query or command to run before each subquery in a queue. (e.g. Query\_Band)

**after\_query [queue-name] [query;]**

Specify a query or command to run after each subquery in a queue.

**after\_sql [ tdbench-command | "delete" ]**

Specify command(s) to execute to evaluate a SQL :retcode or :linecount

**after\_os [ tdbench-command | "delete" ]**

Specify command(s) to execute to evaluate an OS :retcode

**Test Control Commands**

**at [queue-name] [#]{"s" | "m" | "h"}**

Time or relative time to start a queue.

**Debug [ on | off ]**

Suppress RUN, OS, and SQL during testing

**explain [queue-name]**

Adds "explain" to every subquery in a queue.

**finish [queue-name]**

Ensures queue must finish at least once before the test ends.

**limit [queue-name] [#]{"s" | "m" | "h"}**

Maximum time to run a subquery in a queue before aborting.

**pace [queue-name] [#]{"s" | "m" | "h"}**

interval between queries in a queue.

**param [queue-name] [file delimited parameters] {delimiter | tab} {count]**

Provides parameters for subqueries in a queue.

**prepare [queue-name] {data-type-1, ...}**

Provides parameters for subqueries in a queue run using prepared statements.

**rowcount [queue-name]**

Workers in a queue will count all returned rows (affects performance!)

**rows [queue-name] {maximum row count}**

Workers in a queue will returns some or all rows (can affect performance!)

**Queue Preparation Commands**

**replicate [queue-name] [multiplier]**

Duplicates a queue's commands by a multiplicative factor.

**save [queue-name | runid ] [pathname-template-to-save-into]**

Saves a queue's commands in a set of files for future tests or the statement in a runid into a file.

**shuffle [queue-name]**

Reorders a queue's commands and parameters randomly.

**Immediate Commands**

**goto [labelname | testname]**

Skips until LABEL or DEFINE statement with the label/test name specified

[**if**](#_IF,_GOTO_and) **[variable] [ = | != | > | >= | < | <= ] [constant] [“THEN” statement]**

Tests results of prior OS, SQL or RUNs to determine next statement, test name or label to process

**label [labelname]**

Provides a name to be referenced by GOTO

**OS { file:fileame | file:null } [command-to-execute]**

Immediately executes an operating system command

**sql { file:filename | file:null | text:filename} [db-alias-name] [sql-command";" | filename-of-sql-commands]**

Immediately executes one or more SQL statement

**Variables:**

| **Type** | **Examples** |
| --- | --- |
| **INCLUDE or EXEC**  put in file being run | :i1 :i2  :i3 … |
| **Environment** May be different in Windows or Linux. Use SET command in Windows or export in Linux to set | ${HOSTNAME}  ${LOGNAME) ${PWD} ${USERNAME} ${MyPass} ${scale} |
| **Program Variables** Use in script during or after test. Use TdBench  echo :all for complete list | :runid :testname  :runsecs  :resultcount  :linecnt  :retcode |
| **PARAM variables** from each delimited field in PARAM file | :1  :2  :3 … |
| **Worker Variables** Use in queries, substituted dynamically during execution | :runid  :testname  :queryname  :queue  :timestamp |

**H2 (Results) Database Commands**

**archive [tables-prefix-name] {“from” other\_prefix} {“where” where-conditions]**

Archives tests in H2 in separate tables. Optionally may copy from another archive and apply where…

**delete [runid]**

Removes rows from TestResults table. Useful when test fails with thousands of errors

**Dump {archive-name} {“to” zip-prefix}**

Writes all test information from H2 archive to CSV files and zips as archive-name.zip or zip-prefix.zip.

**list {archive-name} {-count | -recent-count | runid {to runid] | “where” where-conditions}**

Lists all tests captured in the H2 database.

**note [runid] {add} {notes or observations to set, add, or omit to delete note}**

Adds a note to a RunID describing conditions or usefulness of test to final results

**remove [tables-prefix-name]**

Deletes archives data in H2

**Restore [archive-prefix] {“to” other-archive-prefix}**

Restores tests archived/dumped from a zip file to tables with archive-prefix or other-archive-prefix

**select [h2 SQL select query]**

Allows query against test results output to console (alias for SQL H2 SELECT...).

**H2 Tables**

**TESTTRACKING … *the version inside TdBench’s H2 database***

RUNID

TESTNAME

TESTDESCRIPTION

STARTTIME

STOPTIME

LOGCOUNT

RESULTCOUNT

ERRORCOUNT

NOTES

URL

**TESTTRACKINGLOG table in H2**

RUNID

LOGLINENO

LOGTIMESTAMP

LOGTYPE … Input, Info, or Error

LOGTEXT

**TESTRESULTS**

RUNID

QUERYID

SUBQUERYID

QUEUENAME

WORKERID

USERNAME

QUERYNAME

RETURNCODE

ERRORMSG

STARTTIMESTAMP

RESPTIMESTAMP

RESPTIMEMS

ROWS

DEFICITMS

PARAMS

**H2 Analysis Views**

**RPT\_TESTS**

Summarizes at the RunID level.

|  |  |
| --- | --- |
| RUNID | *group by* |
| TESTNAME | *group by* |
| STARTTIME | *group by* |
| STOPTIME | *group by* |
| RUNSECS | *group by* |
| NUMEXEC | *calculated* |
| NUMNOERROR | *calculated* |
| NUMQUEUES | *calculated* |
| NUMWORKER | *calculated* |
| AVERESP | *calculated* |

**RPT\_QUEUES**

Summarizes by queue within each RunID

|  |  |
| --- | --- |
| RUNID | *group by* |
| TESTNAME | *group by* |
| QUEUENAME | *group by* |
| NUMWORKER | *calculated* |
| NUMEXEC | *calculated* |
| NUMNOERROR | *calculated* |
| AVERESP | *calculated* |
| MINRESP | *calculated* |
| MAXRESP | *calculated* |

**RPT\_QUERIES**

Summarizes by query name within queue within each RunID.

|  |  |
| --- | --- |
| RUNID | *group by* |
| TESTNAME | *group by* |
| QUEUENAME | *group by* |
| QUERYNAME | *group by* |
| NUMQUERY | *calculated* |
| NUMEXEC | *calculated* |
| NUMNOERROR | *calculated* |
| AVERESP | *calculated* |
| MINRESP | *calculated* |
| MAXRESP | *calculated* |

**RPT ERRORS**

Summarizes errors by RunID, queue, query

|  |  |
| --- | --- |
| RUNID | *group by* |
| TESTNAME | *group by* |
| QUEUENAME | *group by* |
| QUERYNAME | *group by* |
| RETURNCODE | *group by* |
| ERRORMSG | *group by* |
| NUMERRORS | *calculated* |